

样例程序：SVPWM + V/F

1. 本程序实现的功能如下：

- 配置 CAPCOM6。
- 使用 CC60,CC61,CC62,COOUT60,COOUT61,COOUT62 作为 PWM 输出，工作于互补输出模式。
- 在 T3 定时中断中，使各个通道的占空比按照 SVPWM 规律变化。

使用到的模块：CAPCOM6、GPT1。

关于 SVPWM 生成的原理请参考

<http://www.infineon.com/cgi-bin/ifx/portal/ep/programView.do?channelId=64397&programId=35835&programPage=%2Fep%2Fprogram%2Fdocument.jsp&pageTypeId=17099>

中 16 Bit CMOS Microcontroller Product XC164CS - Space Vector Modulation 的说明。

2. 模块介绍

2. 1 CAPCOM6

XC164CM 提供功能强大的 PWM 比较捕获单元 CAPCOM6,内部包含 2 路 PWM 定时器，T12 模块提供 3 对比较/捕捉通道，T13 提供 1 路 PWM 输出，使 XC164CM 适合各种交流电机或逆变器的控制。内部集成适用于 BLDC 控制的霍尔信号、反电动势检测功能。此外还包括用于多项电机控制的块交换模式。

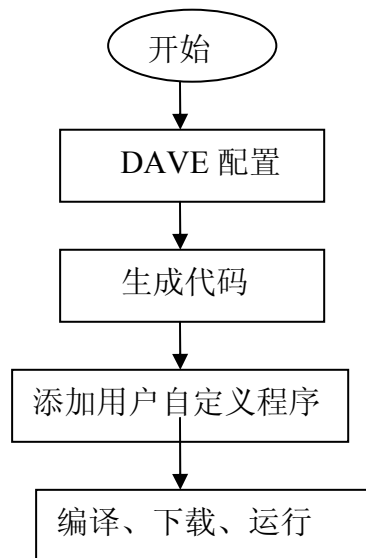
关于 CAPCOM6 模块功能的详细介绍，请参照 XC164CM 用户手册。

2. 2 GPT1

XC164CM 提供了两个通用定时器模块：GPT1，GPT2。GPT1 内部有 3 个定时器（T2、T3、T4），GPT2 内部有 2 个定时器（T5，T6）。GPT1 内部 3 个定时器可以独立使用，也可以配合使用，可工作于定时器，计数器，QEP 输入（正交编码器输入）。GPT2 内部的两个定时器可以独立使用也可以配合使用。具有灵活的工作模式。

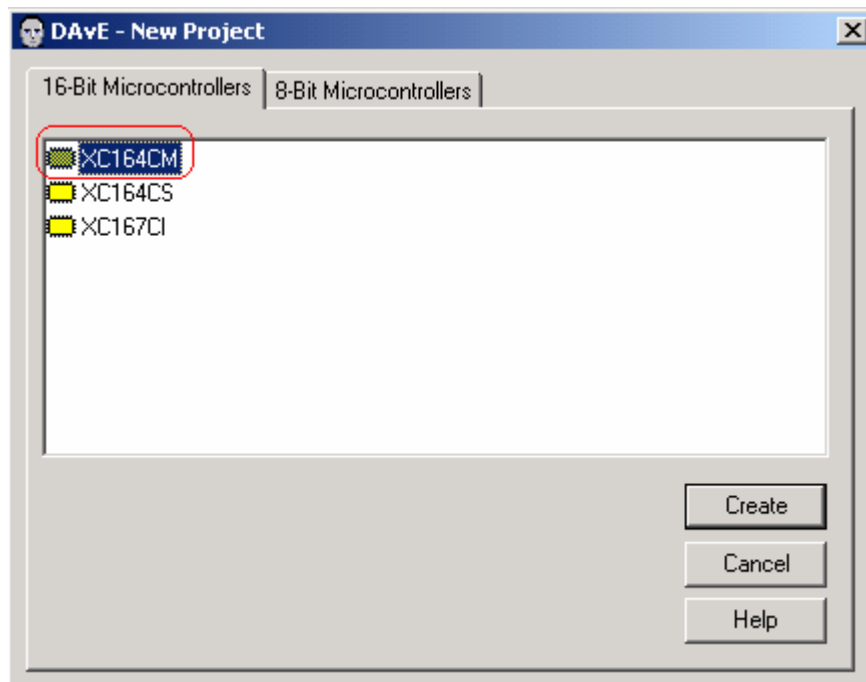
关于 GPT1、GPT2 模块功能的详细介绍，请参照 XC164CM 用户手册。

3. 操作流程

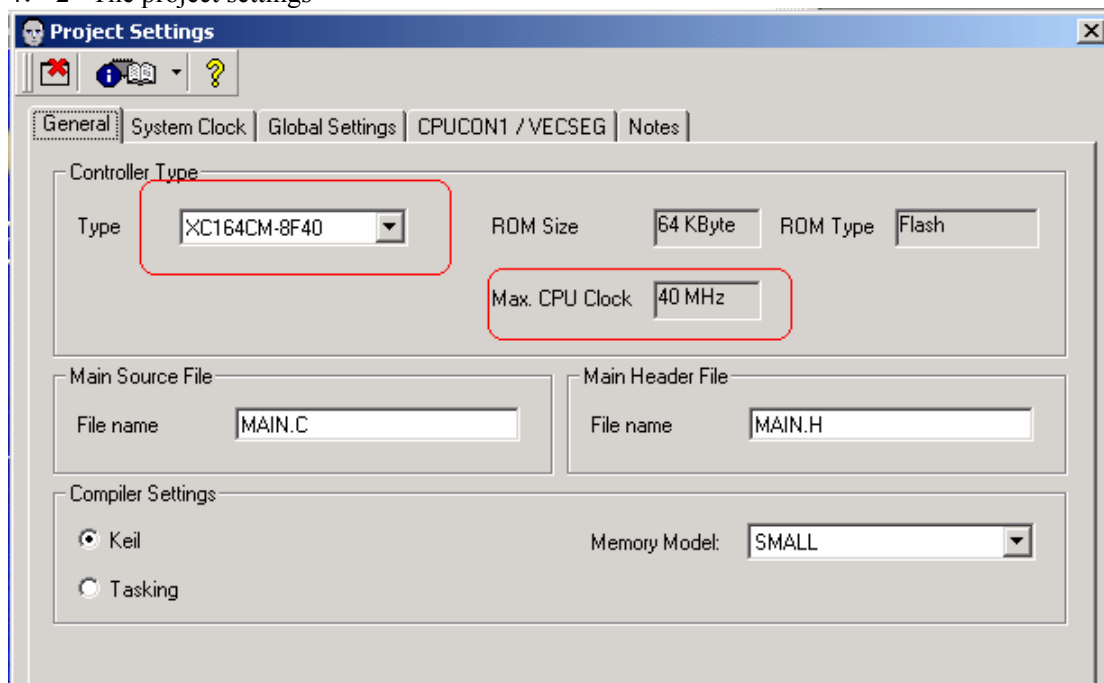


4. DAVE 配置

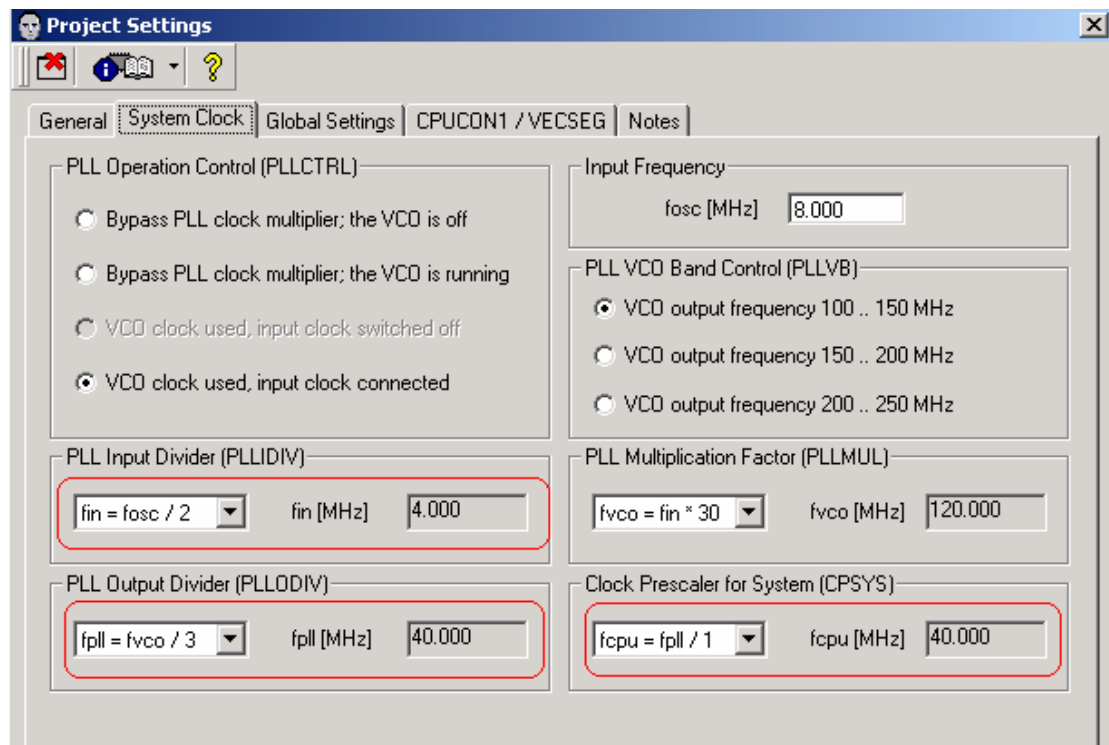
4. 1 New project: select XC164cm,



4. 2 The project settings

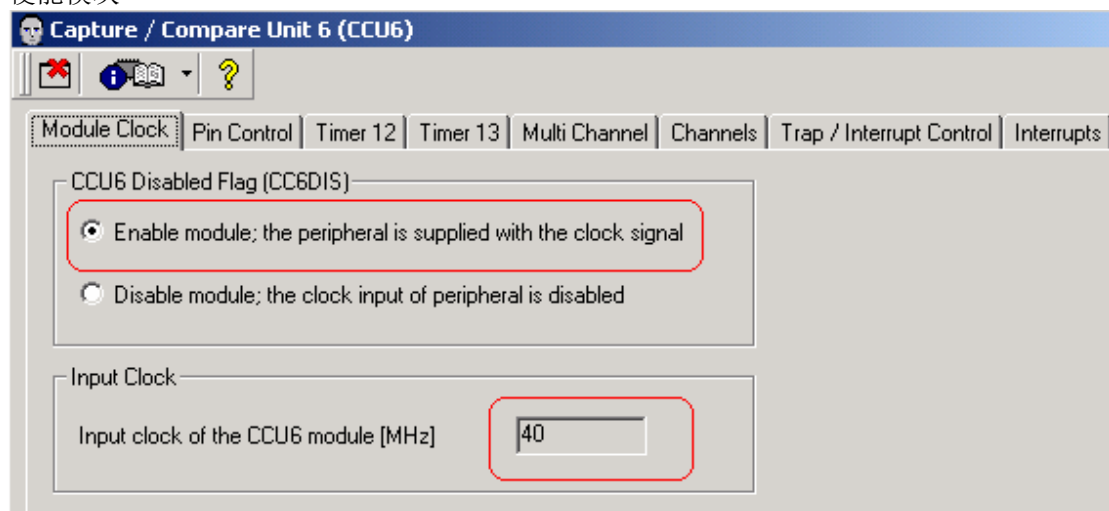


System clock

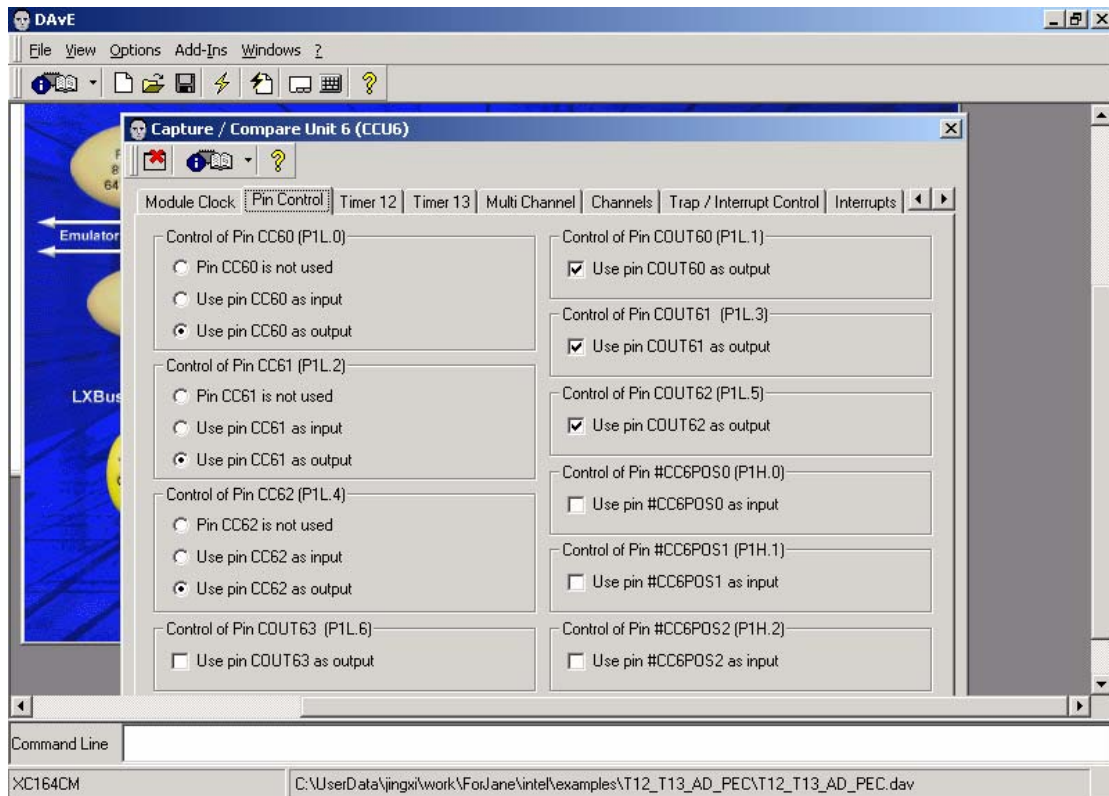


4. 3 配置 CAPCOM6

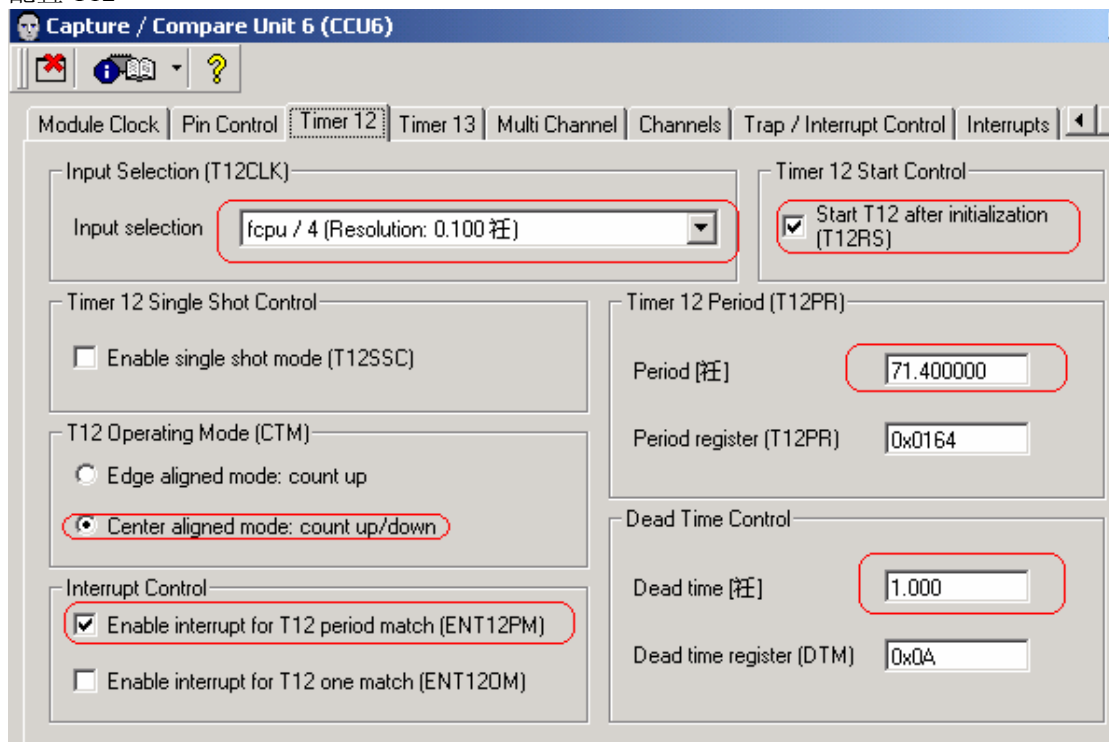
使能模块



配置 CC60、CC61、CC62、COUT60、COUT61、COUT62 为输出



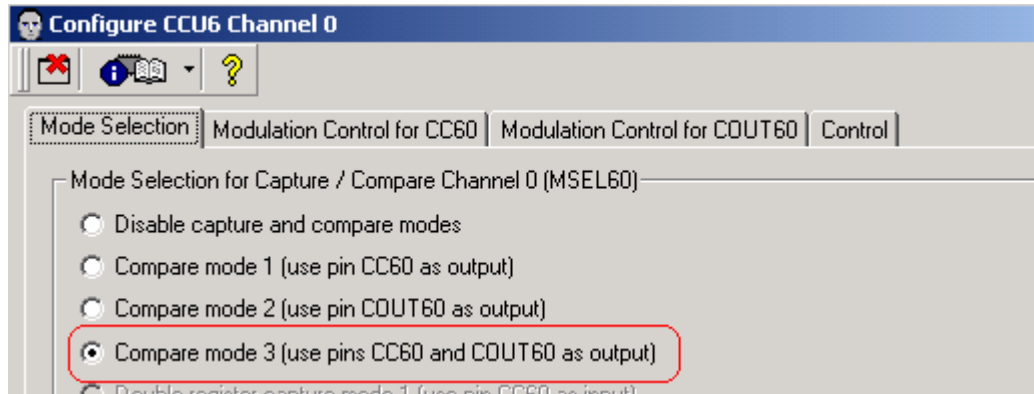
配置 T12



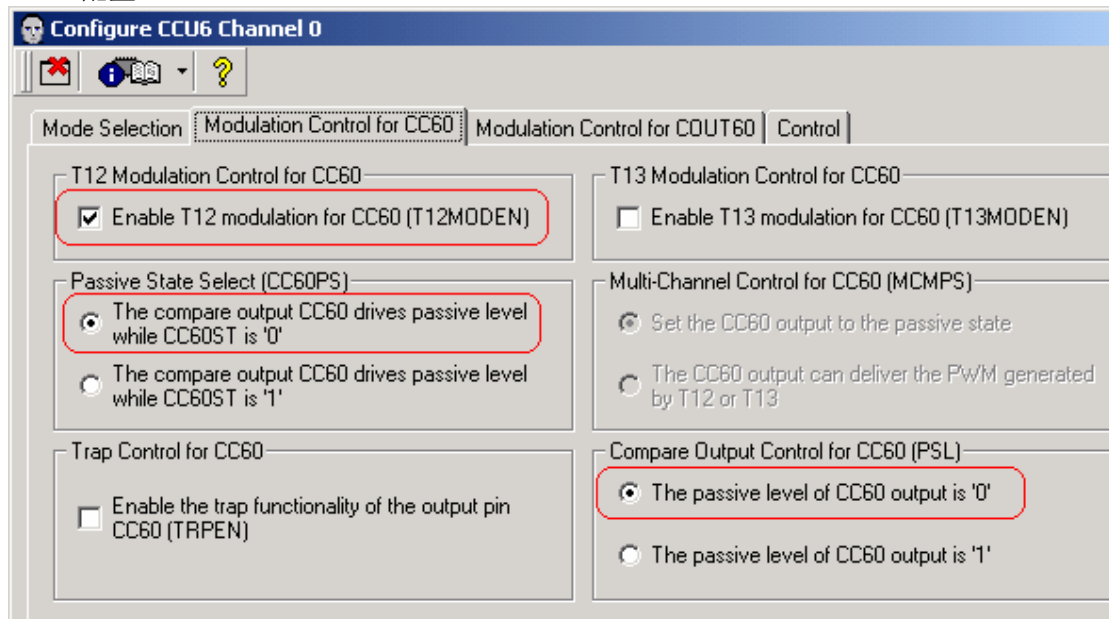
配置各个通道

选择 compare mode 3, T12 modulation, deadtime generation, duty cycle=50%,

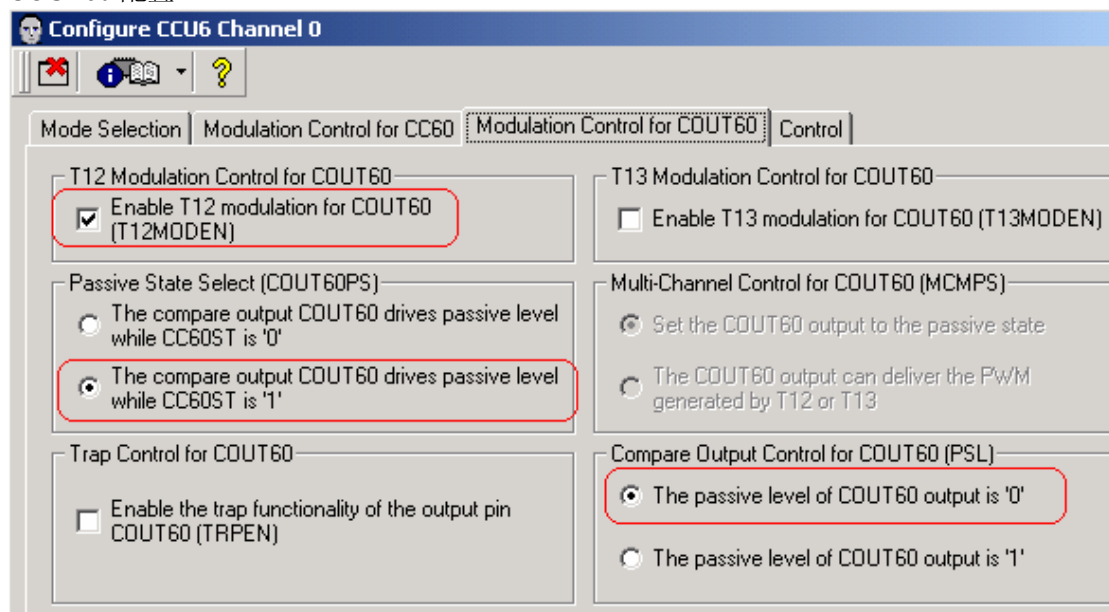
以通道 0 为例：



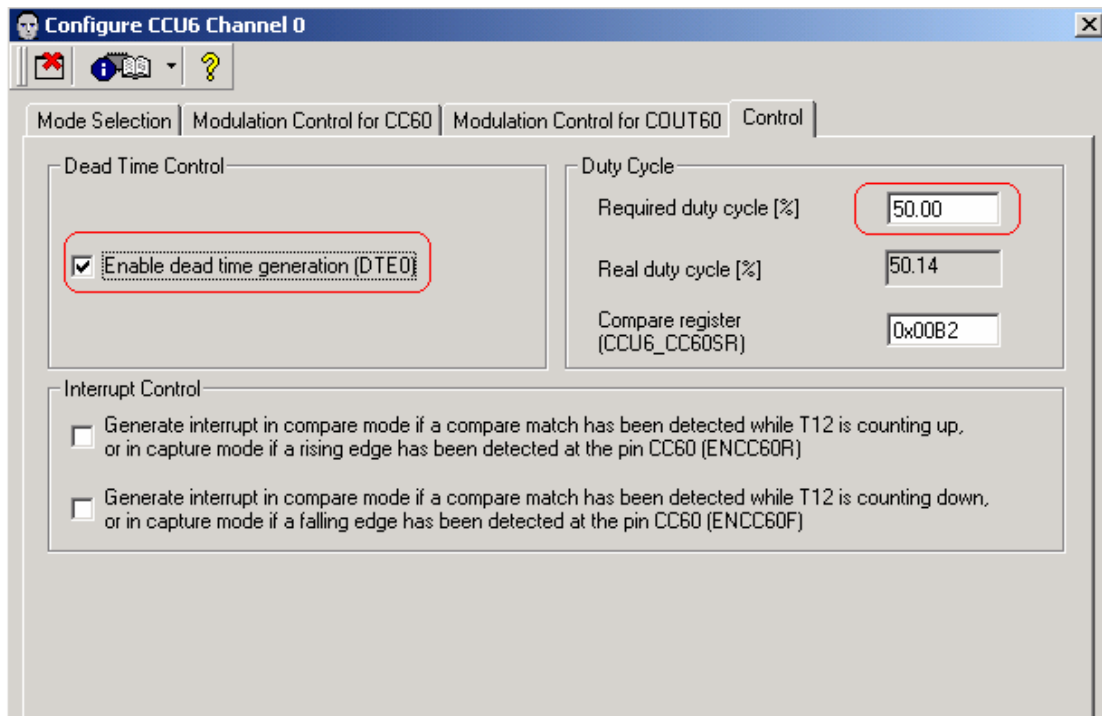
CC60 配置



COUT60 配置

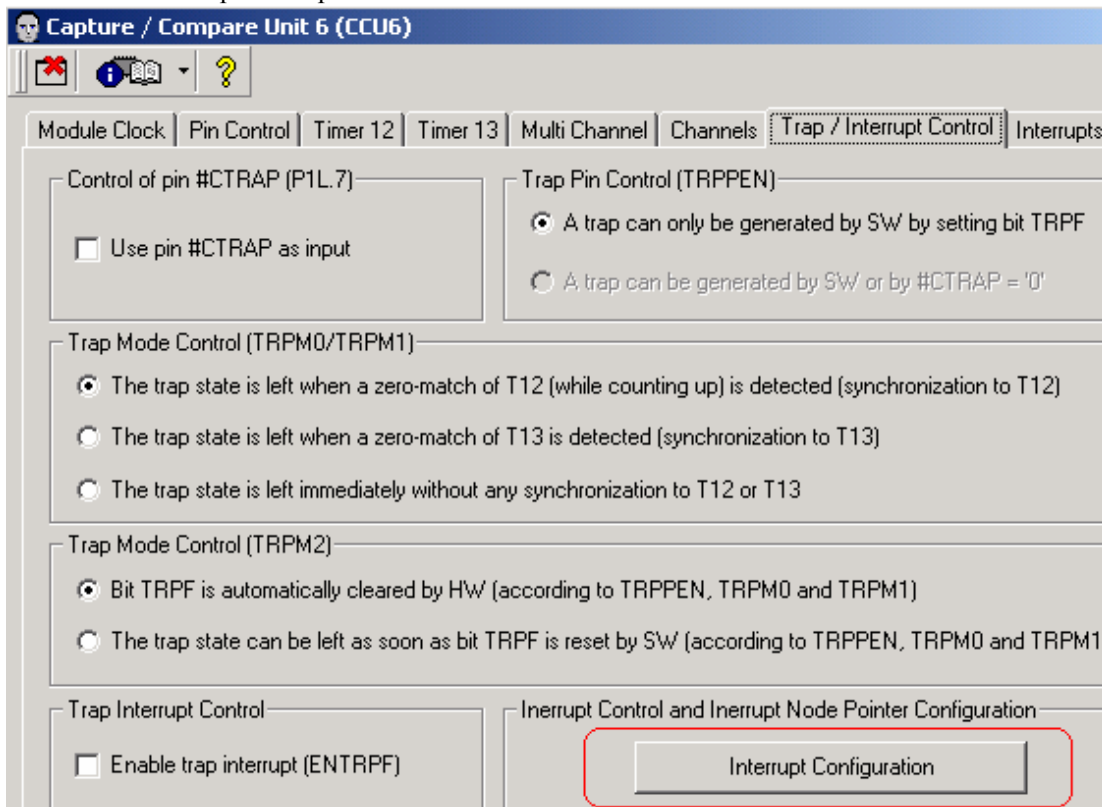


占空比及死区时间配置

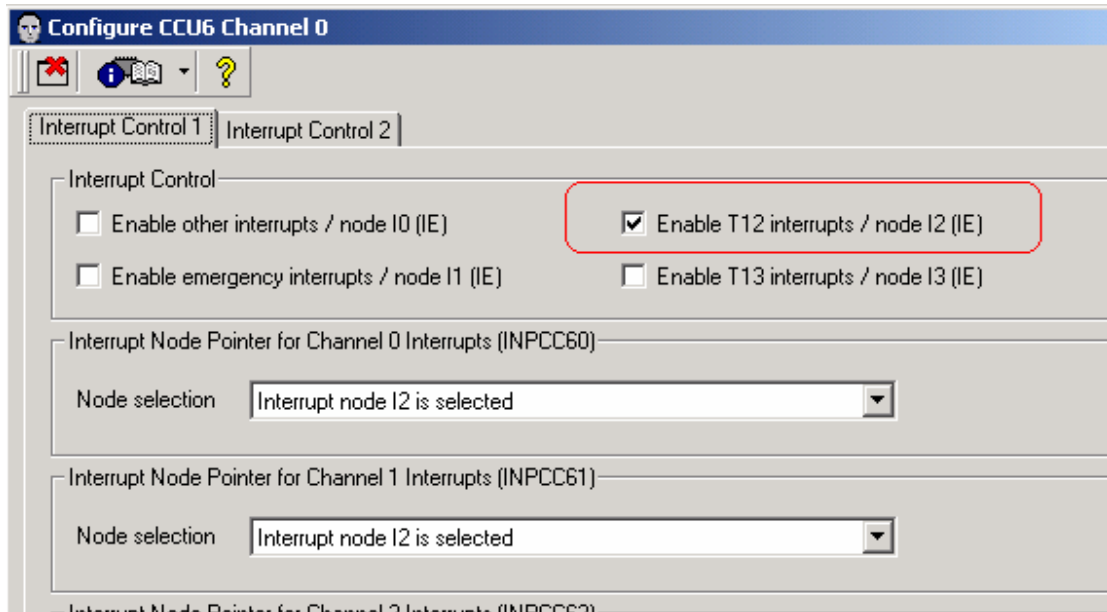


通道 1、2 配置相同。

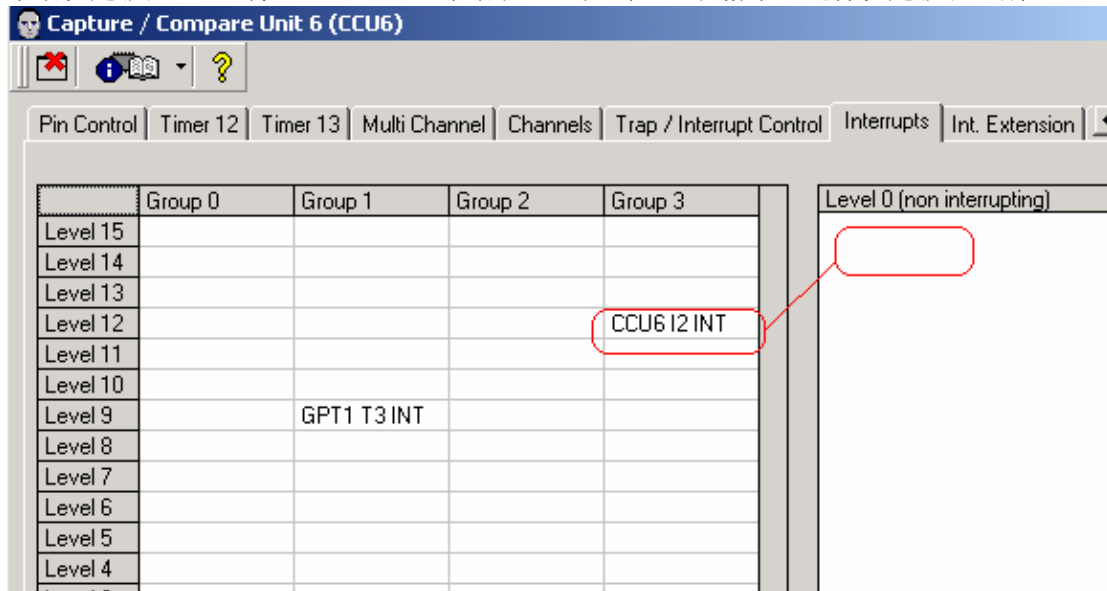
配置中断，在 Trap/Interrupt 页面



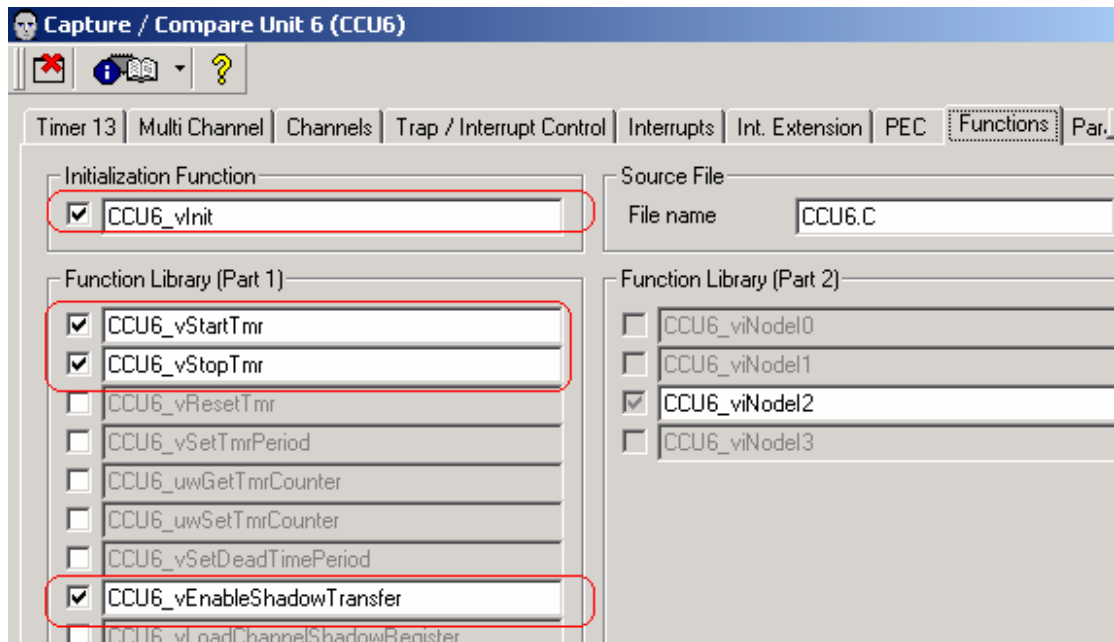
Enable T12 node



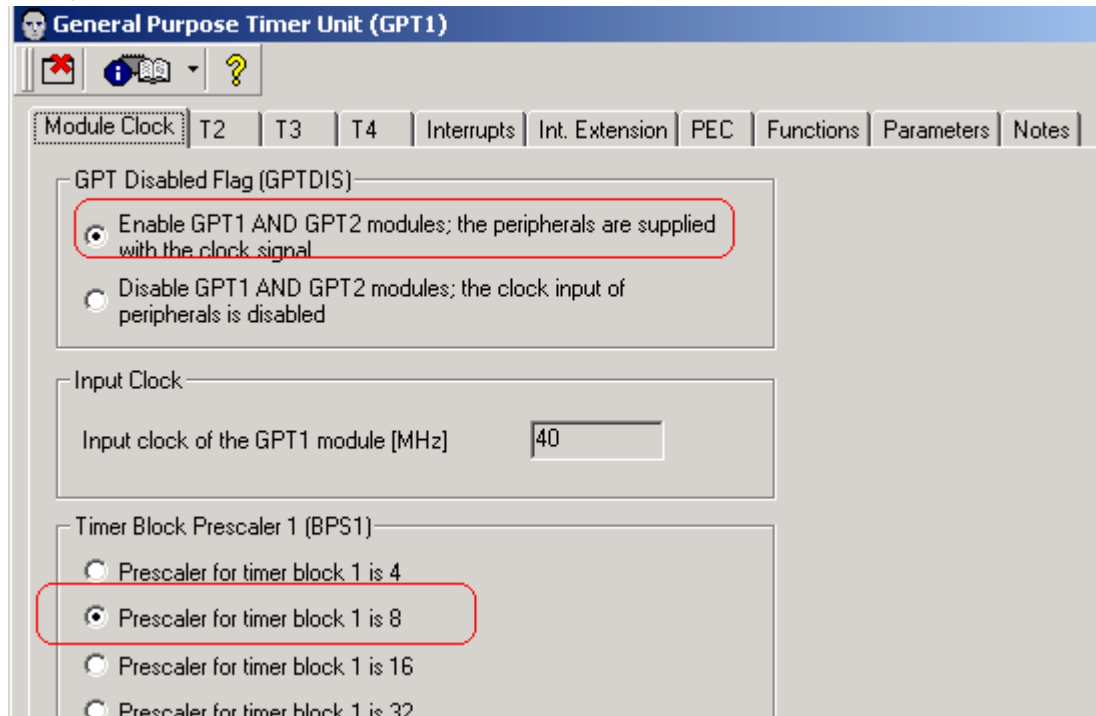
中断优先级配置：将 CCU6 I2 INT 中断从右边拖到左边表格中。选择优先级和组别。



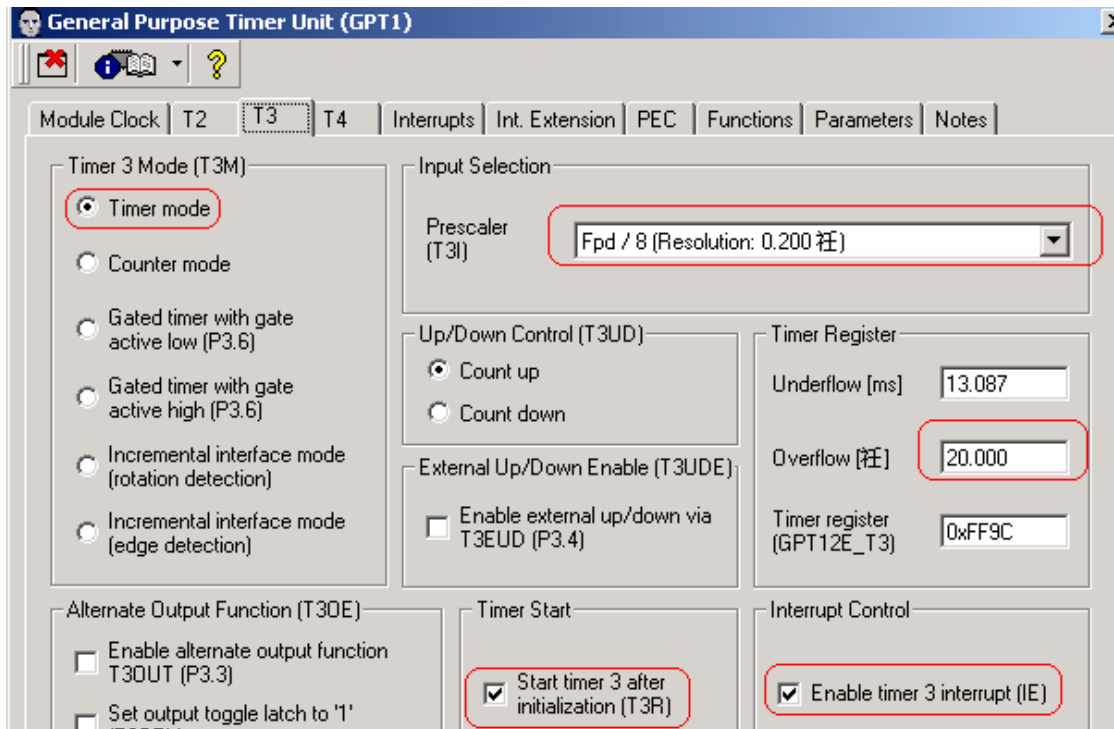
在 functions 页面，选择 CCU6_vInit,生成 ccu6.c 文件。



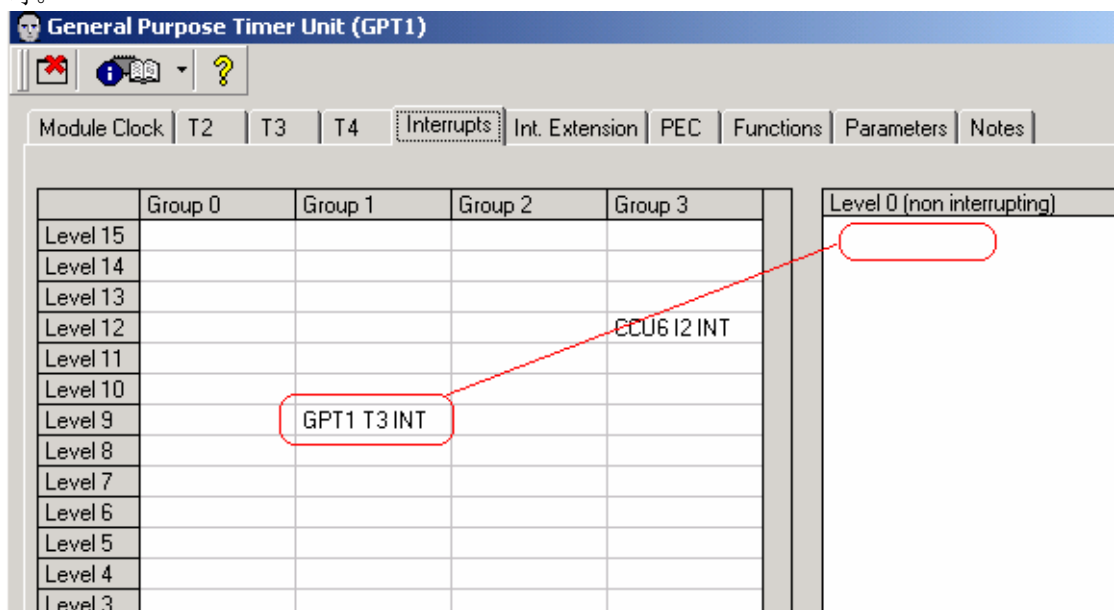
4. 4 配置 GPT1



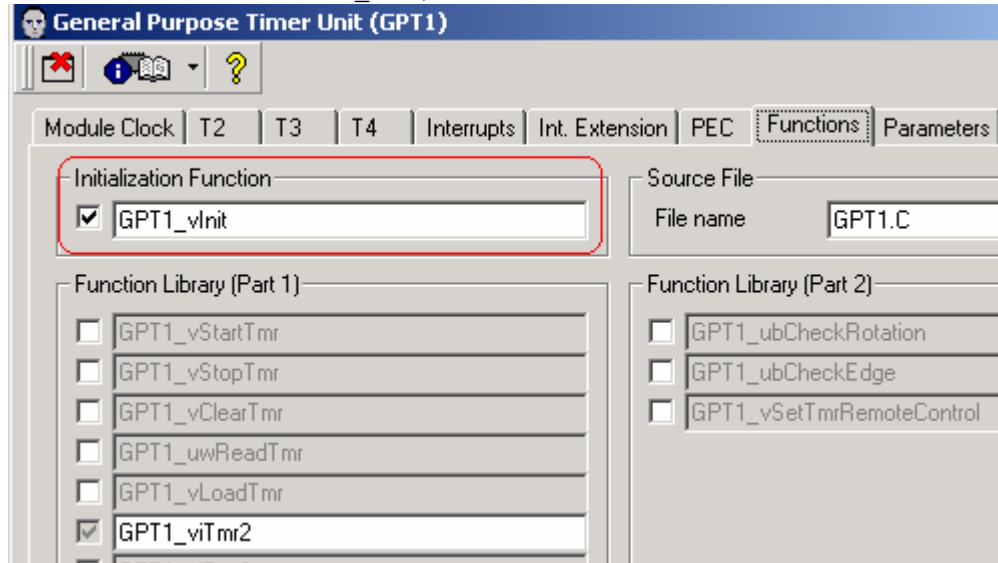
产生简单的定时中断，使用 T3,选择 enable timer 3 interrupt, start after initialization.



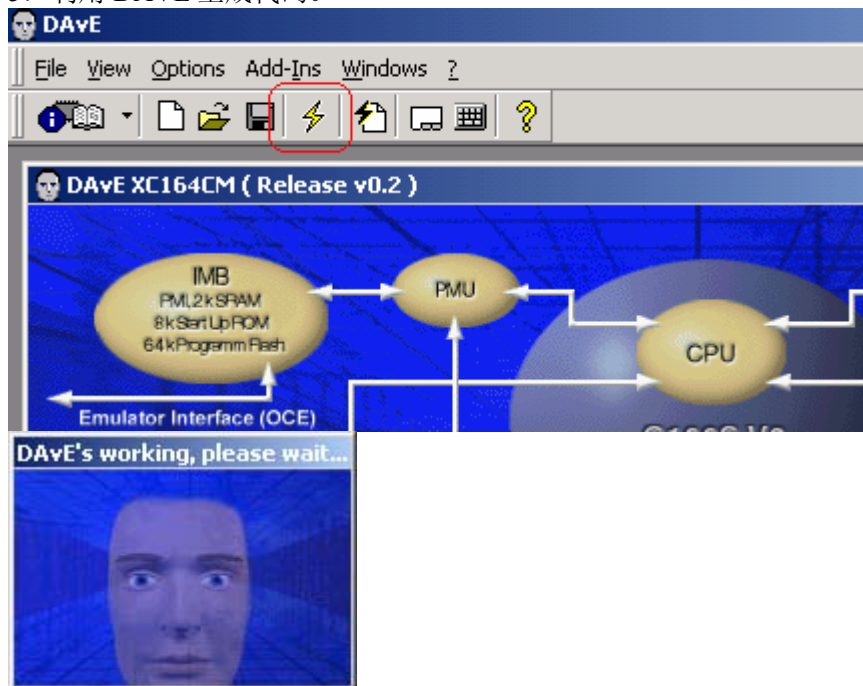
在中断页面, 设置 GPT1 T2INT 的中断优先级和组别。从右边拖到左边表格中相应的位置即可。



在 functions 页面: 选中 GPT1_vInit,生成 GPT1.c 文件。



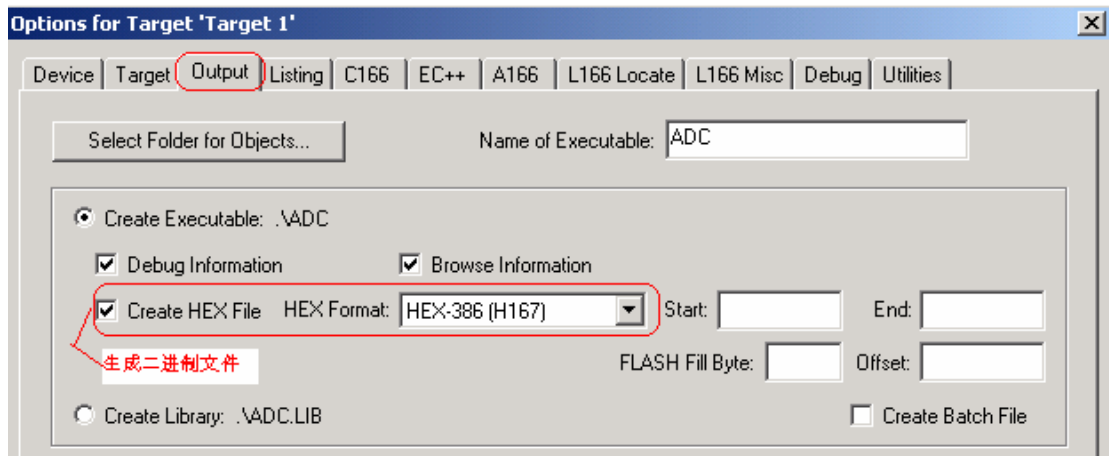
5. 利用 DAVE 生成代码。



6. 修改用户代码

6. 1 生成 uVision 工程文件

做完以上步骤之后工程文件夹中会出现 keil 图标的 dpt 文件，双击进入 keil 环境。第一次进入 keil 环境需要设置：project—options for target 'target 1'。如下所示：



6. 2 Main.c

添加 while(1); 添加变量及引用变量声明

```
extern const unsigned int PWMFrequency[15]; // T12 周期值 1k - 15k
unsigned int TPWM; // 0x03E7 // for 0.1ms
```

```
void main(void)
```

```
{
// USER CODE BEGIN (Main,2)
```

```
// USER CODE END
```

```
MAIN_vInit();
```

```
// USER CODE BEGIN (Main,4)
```

```
TPWM = PWMFrequency[9]; // 选择载波频率。
```

```
while(1); // 添加 while(1)。
```

```
// USER CODE END。
```

6. 3 Main.h

添加 v/f 控制的预编译控制

```
#ifndef _V_F_CONTROL_
#define _V_F_CONTROL_
#endif
```

6. 4 将例程中的 SVM.c、table.c、table.h、tableQ15.h、def.h 拷贝到项目所在文件夹。

SVM.c 与 svpwm 生成相关的函数。

关于具体 pwm 生成请参考

<http://www.infineon.com/cgi-bin/ifx/portal/ep/programView.do?channelId=-64397&programId=35835&programPage=%2Fep%2Fprogram%2Fdocument.jsp&pageTypeId=17099> 中 16 Bit CMOS Microcontroller Product XC164CS - Space Vector Modulation 的说明。

Table.c

PWMFrequency: 定义了 1 k-15k 载波对应的 T12 定时器值。

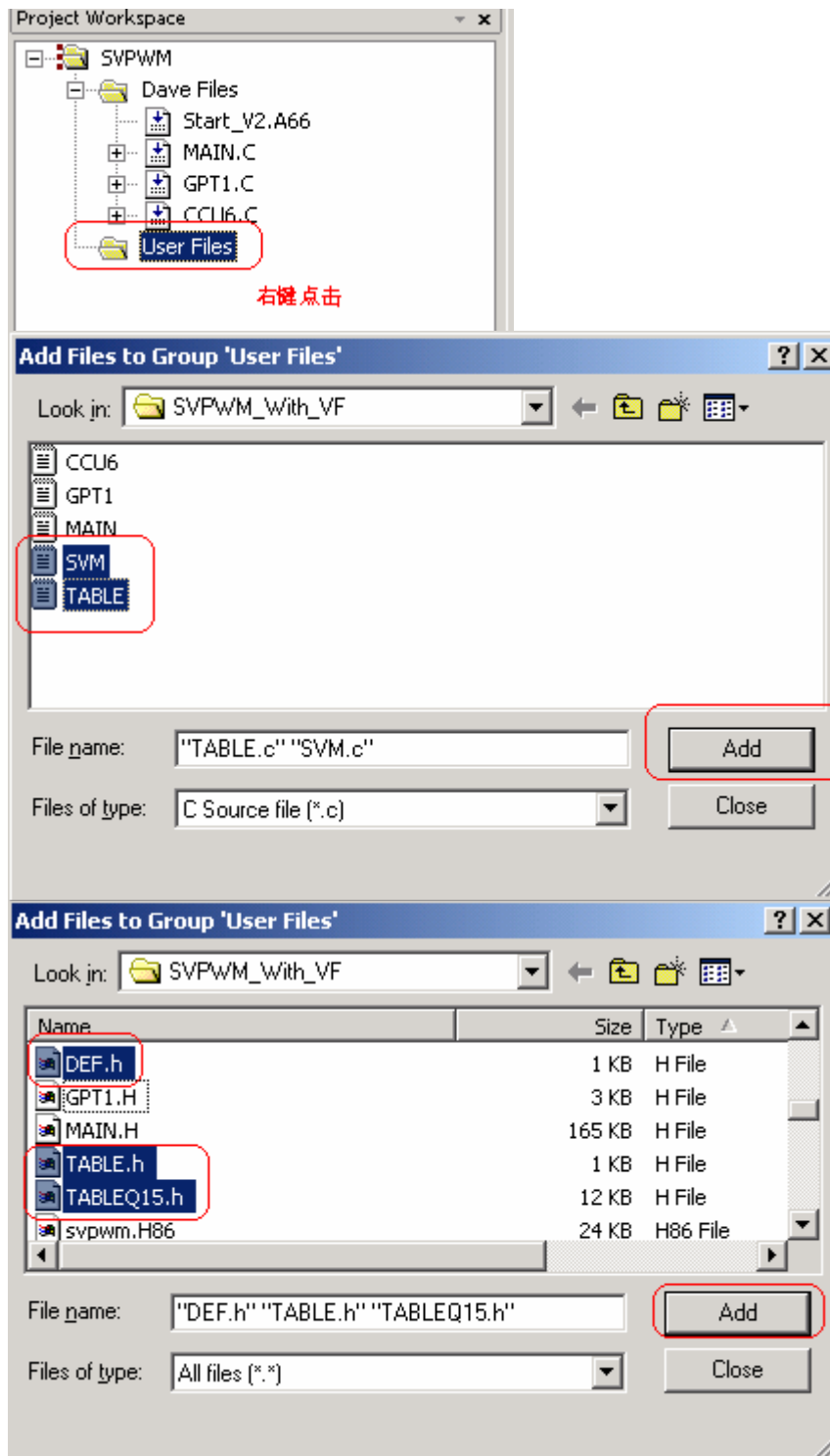
sin_table[], cos_table[]: 正弦、余弦表。

Table.h 与 Table.c 对应的.h 文件

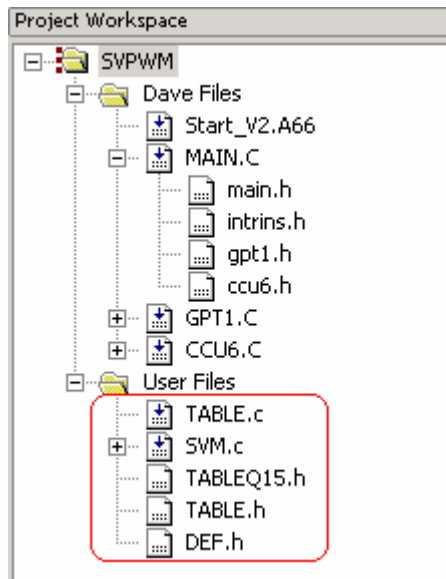
tableQ15.h 定义了与 svpwm 相关的 1Q15 模式的一些数组。

Def.h 定义了与 svpwm 相关的常量及数据结构。

6. 5 向项目中添加 SVM.c、table.h、tableQ15.h、def.h。



添加完后如下所示:



6. 6 CCU6.c

在 T12 定时中断中添加产生 SVPWM 相关的程序

引用文件声明:

```
#include "DEF.h"
```

引用变量声明

```
extern const int sin_table[1001];
extern const int cos_table[1001];
extern unsigned int TPWM;
```

定义全局变量:

```
TComplex StatorCurrentQ15; // 1Q15 格式的定子电流。
unsigned int angle; // 角度值，由 T3 定时中断更新。
```

引用外部函数声明:

```
extern void SVM(TComplex *m, char *Sector);
```

```
void CCU6_viNodeI2(void) interrupt CCU6_NodeI2_INT
{
    // USER CODE BEGIN (NodeI2,2)

    // USER CODE END

    if(CCU6_ISR & 0x0080) // if CCU6_ISR_T12PM
    {
        // timer T12 period match detection

        // USER CODE BEGIN (NodeI2,19)

        // 利用 T3 模拟的角度信息，产生 SVPWM。
        StatorCurrentQ15.real = cos_table[angle];
        StatorCurrentQ15.imag = sin_table[angle];
        CCU6_T12PR = TPWM; // load CCU6 T12 period register
        SVM(&StatorCurrentQ15,&Sector);
        // USER CODE END

        CCU6_ISR = 0x0080; // clear flag CCU6_ISR_T12PM
    }
}
```

```
} // End of function CCU6_viNodeI2
```

6. 7 GPT1.c

引用外部变量

```
extern unsigned int angle; // 角度
```

定义全局变量:

```
unsigned int T3Count; // 软件计数
```

// 针对 v/f 控制的预编译

```
#ifndef _V_F_CONTROL_
```

```
unsigned int V_F_Ratio; // V/F 调制深度。
```

```
#define VF_Con_VFRATIO 1310; // V/F 比率，需根据实际修改。
```

```
#endif
```

```
void GPT1_viTmr3(void) interrupt T3INT // T3 定时中断
```

```
{
```

```
    // 软件重置 T3 的定时值。
```

```
    GPT12E_T3CON_T3R = 0; // set timer 3 run bit
```

```
    GPT12E_T3 = 0xFF9C; // load timer 3 register
```

```
    GPT12E_T3CON_T3R = 1; // set timer 3 run bit
```

```
    T3Count++; //更新软件计数值
```

```
    angle = T3Count * Freq / 50; // 计算角度信息
```

```
    // V/F 控制时根据频率值计算调制深度。
```

```
    #ifndef _V_F_CONTROL_
```

```
    V_F_Ratio = Freq * VF_Con_VFRATIO;
```

```
    #endif
```

```
    // 当每一频率周期完成后，重置计数值，并增加频率，最大频率 50Hz。
```

```
    if(T3Count == 50000 / Freq)
```

```
    {
```

```
        T3Count = 0;
```

```
        Freq ++;
```


```
        if(Freq >= 50)
```

```
            Freq = 50;
```

```
    }
```

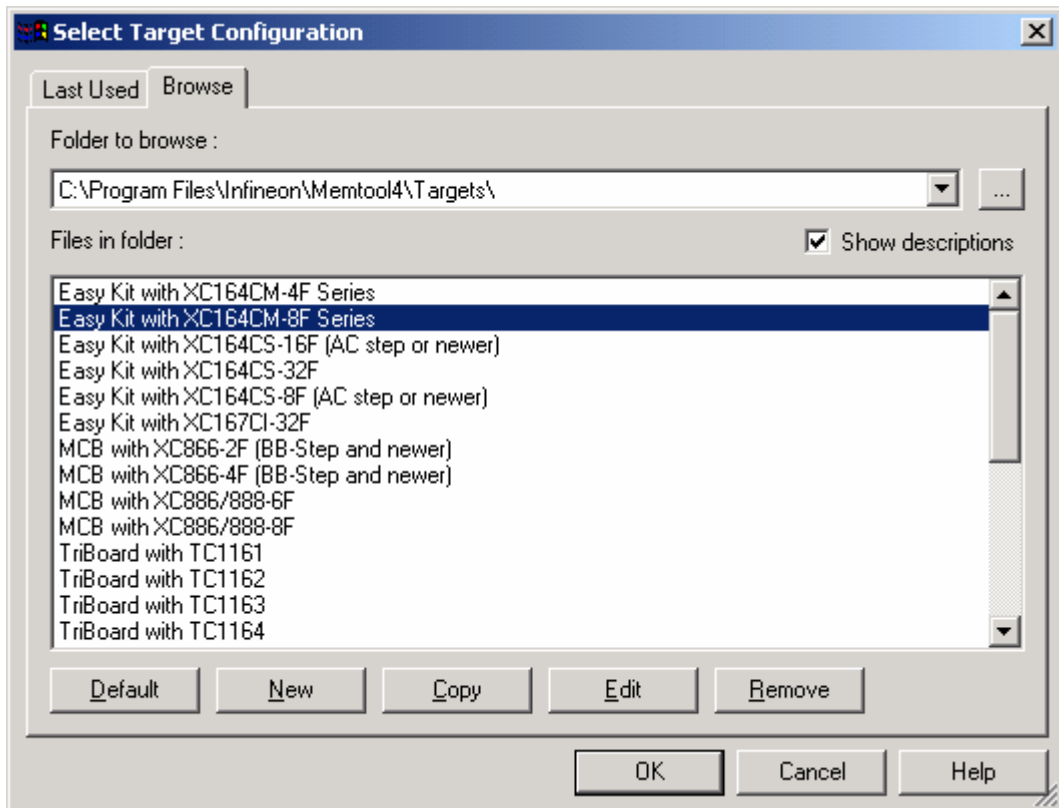
```
}
```

7. 编译

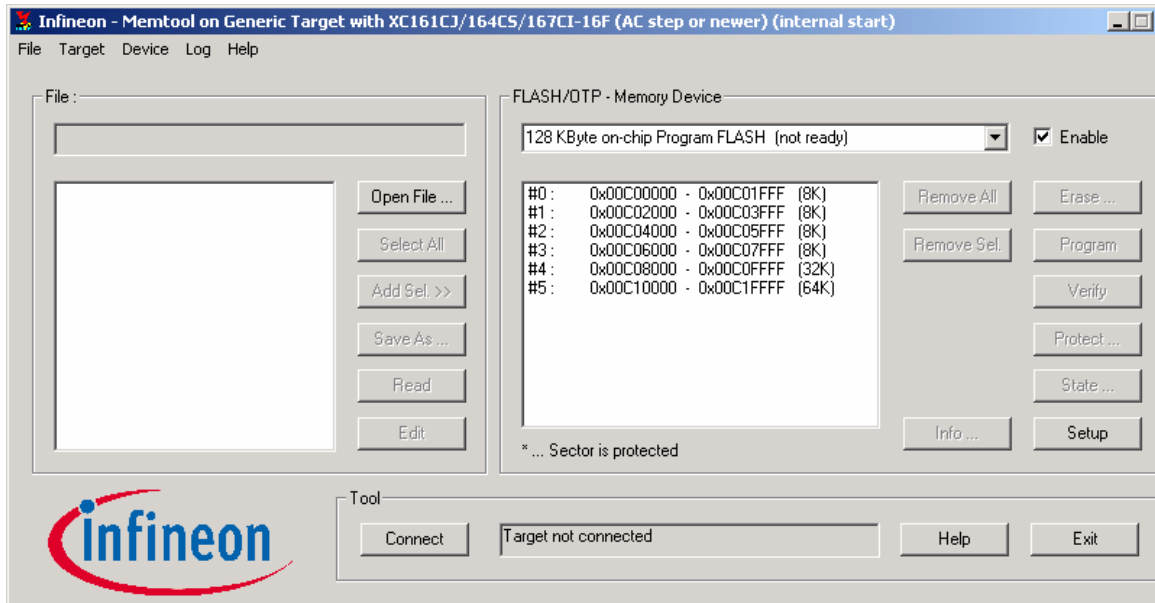
点击  图标进行编译连接。如有错误进行更改，直到出现‘0 Errors found.’。

8. 下载

利用 memtool 软件将上面生成的 h86 文件下载到单片机。打开 memtool 软件，点击菜单 Target—Change，选择 XC164CM-8F。界面如下：

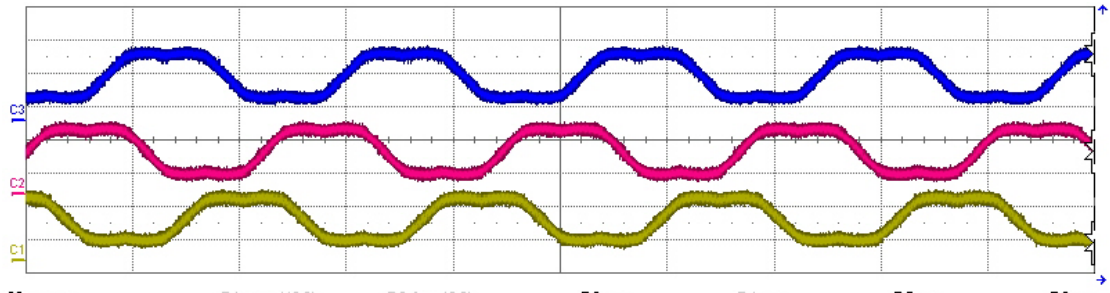


击 OK 出现如下对话框。



点击 ‘connect’ 进行通讯连接。通讯成功之后，按照顺序 open file...—select all—add sel.>>将 h86 文件添加到右边框中，然后选择 ‘Erase...’ 和 ‘Program’ 进行擦除、编程。如有必要可点击 ‘Verify’ 进行校验。

9. 运行
波形如下：



Measure	P1:ampl(C3)	P2:freq(C3)	P3:---	P4:---	P5:---	P6:---
value			---	---	---	---
status						

C1	C2	C3
2.00 V/div	2.00 V/div	2.00 V/div
-7.204 V ofst	-3.200 V ofst	1.202 V offset
↓ 1.10 V	↓ 2.57 V	↓ 3.98 V
↑ 1.10 V	↑ 2.57 V	↑ 3.98 V

Tbase	100.0 ms	Trigger	25
	10.0 ms/div	Auto	7.00 V
250 kS	2.5 MS/s	Edge	Positive
X1=	-50.0004 ms	ΔX=	0.0 μs
X2=	-50.0004 ms	1/ΔX=	---